Speedy and Scalable File-System Benchmarking with *Compressions*

Nitin Agrawal (student), Andrea C. Arpaci-Dusseau and Remzi H. Arpaci-Dusseau
Department of Computer Sciences, University of Wisconsin-Madison
{nitina, dusseau, remzi}@cs.wisc.edu

Everyone cares about data. Thus, it is important to accurately assess the performance of the file and storage systems that store our valuable data. As designers and evaluators of these systems, it is our responsibility to ensure that they are benchmarked under real-world conditions with a realistic choice of workloads.

Unfortunately, file and storage systems are currently difficult to benchmark. There are two important reasons for this. First, recreating real-world conditions for the purposes of benchmarking is hard. Second, constructing realistic synthetic benchmarks that can be used as substitutes for real and complex workloads is also a hard problem.

These reasons point to a common problem – in the world of benchmarking, ease of use and portability dominate over realism and accuracy. More realistic workloads (and realistic configurations of these workloads) tend to be larger and more complex to set up. File system traces (*e.g.*, from HP Labs [3]) are good examples of such workloads, often being large and unwieldy.

Two trends further exacerbate the difficulty in file and storage benchmarking. First, storage capacities have seen a tremendous increase in the past few years; terabyte-sized disks are now easily available for desktop computers; enterprises are now frequently dealing with petabyte-scale storage. Second, popular user applications are taking increasingly longer to run on large disks. Example applications include fsck and desktop search, both of which can take anywhere from several hours to a few days to run on a Terabyte-sized partition. Benchmarking such applications on a large partition is a source of frustration for system evaluators, but running toy workloads on small disks is no longer a reasonable choice. Thus, making it practical to run realistic benchmarks on large disks is an important challenge for the storage community.

In this work we address this challenge by building *Compressions*, a system that enables an evaluator to run the same large benchmark but on a much smaller disk size while also being faster in total runtime. They key idea is to create a "compressed" version of the original file-system image on disk for the purposes of benchmarking. The compressed image differs from the original in two distinct ways. First, user data blocks are not written out to disk. Second, file system metadata blocks (*i.e.*, blocks belonging to inodes, directories, indirect blocks etc) are laid out more efficiently on disk. In the simplest case, metadata blocks are written out sequentially on the disk.

Compressions uses a modified in-kernel implementation of the Impressions framework [1] to produce data and metadata such that applications remain blissfully ignorant of the fact that they are running on a compressed disk image. Since the layout of data and metadata blocks on disk is now synthetically compressed, the runtime as measured by an application running on Compressions is not indicative of its true performance. In order to deliver accurate performance results for benchmarking, Compressions uses an in-kernel disk model to determine the time each individual request would have taken when running on the original uncompressed image.

Depending on the workload and the underlying file-system image, the size of the compressed image can be anywhere between 1 to 10% of the original, which is a huge reduction in the required disk size for benchmarking. Compressions also reduces the time taken to run the benchmark by avoiding a significant fraction of disk I/O and seeks as observed during preliminary results with the PostMark benchmark [2]. In this talk we will focus on the key components of Compressions, namely the generation of the compressed image and the use of in-kernel Impressions to synthetically generate file data in real time.

In summary, Compressions allows one to benchmark workloads that were once prohibitive to setup and required large disk partitions to run on. The required disk size under Compressions can be orders of magnitude smaller than the original while also allowing the benchmark to execute much faster. Finally, Compressions ensures the accuracy of benchmarking results by using a model of the disk to compute the runtime for the original disk.

References

[1] N. Agrawal, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau. Generating Realistic Impressions for File-System Benchmarking. In *Proceedings of the 7th Conference on File and Storage Technologies (FAST '09)*, San Francisco, California, February 2009.

[2] J. Katcher. PostMark: A New File System Benchmark. Technical Report TR-3022, Network Appliance Inc., October 1997.

[3] E. Riedel, M. Kallahalla, and R. Swaminathan. A Framework for Evaluating Storage System Security. In *FAST '02*, pages 14–29, Monterey, CA, January 2002.